EXHIBIT A

Expert Report of Dr. [redacted]

Securities and Exchange Commission v.
Ripple Labs, Inc., Bradley Garlinghouse and Christian A. Larsen

Confidential

October 4, 2021
Updated January 25, 2022

Contents

1 Introduction

1.1 Assignment

I have been engaged by the Securities and Exchange Commission ("SEC"), through [redacted] ("[redacted]") to provide expert testimony in the matter of Securities and Exchange Commission v. Ripple Labs, Inc., Bradley Garlinghouse and Christian A. Larsen, pending in the United States District Court for the Southern District of New York. The SEC has retained me to independently analyze and opine on: (1) whether the distributed ledger system on which XRP token is transacted ("XRP Ledger") is a centralized or a decentralized system as of the date of this report, and (2) what would likely happen to the XRP Ledger if Ripple Labs Inc. ("Ripple") ceased functioning.

Before reaching those questions, SEC asked me to provide answers to certain background questions:

Prefatory Questions:

- *(*P1) Describe the basic operating principles of blockchain technology and explain how its consensus mechanisms work.
- *(*P2) Explain the XRP Ledger consensus mechanism, including the concept of Unique Node Lists ("UNLs").

The SEC then asked me to analyze and opine on the following questions:

Questions for Expert Opinion:

- *(*E1) To what extent is the XRP Ledger centralized or decentralized when compared to generally recognized blockchain protocols such as those used by Bitcoin and Ethereum?
- *(*E2) To what extent have Ripple's efforts been needed to support the proper functioning of the XRP Ledger?
- *(*E3) What risks to the XRP Ledger would or might materialize if Ripple "walked away" or "disappeared"?

1.2 Qualifications

I am a computer scientist with 18 years of specialization in fault-tolerant distributed systems, an area of computer science that is at the core of blockchain and decentralized systems. In particular, my core area of expertise are so-called "Byzantine" fault-tolerant ("BFT") distributed consensus protocols. *Byzantine* here refers to the ability of participants in a distributed system, to deviate from the algorithm prescribed to them (e.g., by being malicious, that is by acting to purposefully attempt to disrupt the functioning of the system). The consensus protocol that underlies the XRP Ledger aspires to be in the category of BFT consensus protocols.

I hold a [redacted] from the [redacted] (1996-2001) and a [redacted] degree from [redacted] in distributed systems (2003-2008). My PhD thesis entitled [redacted]

[redacted], dealt with BFT distributed consensus protocols. Before Bitcoin, BFT consensus was only a rather niche area of research. [redacted]

After my PhD, I was a Postdoctoral researcher at [redacted]. After that, in the period from 2010 to 2014, I worked in academia. [redacted]

I am an author of many research papers and patents which are often cited by other researchers. [redacted]

[redacted]

I respectfully ask you to refer to my enclosed CV for additional details.

I have been retained through [redacted], a forensic data analytics and litigation consulting firm. I am compensated by the SEC via [redacted] at the rate of $700 per hour. My compensation is not dependent on me reaching any specific opinion. Members of [redacted] team also performed work in connection with this report and are compensated at a rate ranging from $235 to $520 per hour.

1.3 Documents Relied Upon

For the analysis of the XRP Ledger protocol, I relied on two papers authored by current and former Ripple employees, the official documentation of the XRP Ledger, as well as on reviewing the code of the XRP Ledger server. These sources are listed in detail in Section 4.1.

Furthermore, the "References" section of this report contains a list of other documents and data sources I relied upon in completing the analysis in this report, including a body of scientific research related to the definition of decentralized systems.

Where appropriate, the data sources are given inline in the text, as a web link, footnote or a citation.[1]

2 Summary of Findings

I reviewed the scientific literature on decentralized systems, with which I was familiar, to establish a methodology for evaluating the extent of decentralization of distributed systems.

- I first adopt the basic definition of a decentralized system, as defined by Troncoso et al. [21], which defines decentralized systems as a subset of distributed systems where multiple authorities (parties) control different system components and no authority is fully trusted by all.
- I then refine this basic definition, with the support of the scientific literature, to identify four main aspects of decentralization: Resilience, Inclusiveness, In-Protocol Incentives, and Governance. I define each of these aspects of decentralization in Section 3.1.

I proceed to explain the inner workings and to evaluate the decentralization levels of the Bitcoin (Sec. 3.2) and Ethereum (Sec. 3.3) blockchains, respectively. I thereby answer Prefatory Question P1 and prepare the ground for answering Expert Question E1 (as defined in Sec. 1.1).

I then turn to analysis and explanation of the XRP Ledger protocol in Section 4, in particular to its concept of Unique Node Lists (UNLs), thereby answering Prefatory Question P2. In my analysis of the XRP Ledger I rely solely on the material which I consider endorsed by Ripple and/or its employees, as listed in Section 4.1.

Finally, in Section 5, I give my expert opinion, answering questions E1, E2 and E3, as stipulated in Section 1.1. Below, I give an overview of these findings.

I answer Question E1 in Section 5.1, where I evaluated the decentralization of the XRP Ledger (i.e., its Resilience, Inclusiveness, In-Protocol Incentives, and Governance aspects) and compared it to the decentralization of the Bitcoin (itself evaluated in Sec. 3.2.3) and Ethereum (Sec. 3.3.1) blockchains. An overview of this comparison is given in Table 1.

In summary, the XRP Ledger has low Resilience as it takes corrupting only a single party to be able to compromise key properties of the system.[2] In fact, as a result of its low Resilience, the XRP Ledger does not satisfy the basic definition of a decentralized system [21], and is, therefore, in my opinion, centralized.

The centralization here stems from the following facts pertaining to the XRP Ledger software, which I will detail later in this report:

1. Participants required for the proper operation of the system (nodes) are "curated" by Ripple for inclusion into a special list, called the dUNL, which is to be understood as a *default Unique Node List*.

[1] Beyond these sources, I further considered the following documents related to this case, none of which I relied on in forming my opinions set forth herein:

- "Submission to the Conference of State Bank Supervisors", submission by Ripple Labs Inc. Bates number RPLI_SEC 0086553.
- Case 1:20-cv-10832-AT Document 46 Filed 02/18/21, 79 pages.
- Case 1:20-cv-10832-AT Document 45 Filed 02/15/21, 9 pages.
- Case 1:20-cv-10832-AT Document 43 Filed 01/29/21, 93 pages.

[2] The number of parties that need to be corrupted to subvert key properties of a distributed system is also sometimes called the Nakamoto coefficient.

Decentralization aspect	Ideal Decentralized System	Bitcoin Blockchain	Ethereum Blockchain (with Proof-of-Work)	XRP Ledger
Nakamoto coefficient (Resilience)	always greater than 1, the higher the better	≥ 4	≥ 3	1
Inclusiveness	yes	yes	yes	no
In-Protocol Incentives	yes	yes	yes	no
Governance (public face)	no	no	yes	yes
Governance (tokens allocated at genesis)	0, the lower the better	0%	61.5% (about 10% owner controlled) of today's supply	100% (all owner controlled)

Table 1: Comparison of the XRP Ledger to the Bitcoin and Ethereum blockchains for key aspects of decentralization defined in the decentralization evaluation methodology of Section 3.1.

2. As of the latest release of the XRP Ledger software, referred to as "rippled v1.7.3", Ripple controls the web domain which hosts the service that provides the dUNL to the XRP Ledger participants. Namely, this dUNL provisioning service is deployed at the address `http://vl.ripple.com`.

3. Participants in the XRP Ledger, who use unmodified code of rippled v1.7.3, periodically refresh their locally referenced UNL, which serves as a local list of "trusted participants", by copying the contents provided by the dUNL provisioning service, i.e., the dUNL controlled by Ripple and disseminated at `http://vl.ripple.com`.

4. The design of the XRP Ledger requires, for correct operation of the protocol, a very large overlap (intersection) across UNLs that individual participants use.

5. Therefore, Ripple's dUNL provisioning service needs to be trusted for correct operation of the system. Otherwise, in the case of an untrusted dUNL provisioning service, it could provide participants with UNLs that do not have sufficient overlap, compromising key properties of the XRP Ledger. This makes it possible for a single authority, namely, Ripple as the dUNL publisher, to subvert key properties of the system. This makes the XRP Ledger, by definition of Troncoso et al. [21], and in my opinion, centralized.

This issue of a centralized dUNL publisher, alone, is in my opinion sufficient to render the XRP Ledger centralized. Nevertheless, I conducted an even more detailed evaluation of the XRP Ledger through the prism of other decentralization aspects. These are summarized below:

- I identified another Resilience vulnerability which makes it possible for a single party to subvert key properties of the system, independent of the centralized dUNL publisher issue. This is detailed later in the report (Appendix B).
- The XRP Ledger does not satisfy Inclusiveness, which, in short, refers to a system which provides equal opportunities to participants (see Section 3.1, for detailed definition). While the XRP Ledger allows any participant to join the system, it treats its participants unequally. This inequality stems,

again, from the existence of a Ripple-curated dUNL, which is, in turn, required for the XRP Ledger to function properly.

- Unlike other compared blockchains, the XRP Ledger does not have In-Protocol Incentives, which are defined, in short, as the existence of software-defined incentives for participants to join the system and which contribute to the decentralization of a blockchain (see Sec. 3.1). In contrast, the XRP Ledger solely relies on out-of-protocol actions of existing participants to incentivize new participants to join the XRP Ledger.

- Finally, the XRP Ledger scores poorly in the Governance aspect. For instance, while an ideal decentralized system should have no public face (representative) and should have not pre-allocated tokens at system's inception, the XRP Ledger sits at the opposite end of the spectrum, having pre-allocated all its tokens to people and organizations which serve or have served as its public face.

To answer the next question, Question E2 from Section 5.2, regarding the role of Ripple's efforts in supporting the proper functioning of the XRP Ledger, I first analyzed the situation as of the time of writing of this report, assuming no further changes to current rippled v1.7.3 code, as the answer depends on the software code. Given the nature of the question, I also analyzed some historical aspects of the system, namely the fraction of validators in the dUNL which Ripple and organizations that received funding from Ripple used to control.

My findings show that, today, Ripple's efforts are needed to maintain components of the XRP Ledger secure from internal and external attacks. These efforts relate primarily to publishing a dUNL, at `https://vl.ripple.com`, in a secure way so that a potential attacker (i.e., a malicious adversary, also called a Byzantine [13] attacker) cannot take control over the dUNL publishing service.

In addition, Ripple needs to ensure that the dUNL is curated and populated only with attested validators, since even a single Byzantine validator, combined with an unreliable network, may subvert key properties of the XRP Ledger— as detailed in Appendix B. For this same reason, Ripple needs to maintain security over the 6 validators it itself controls out of 41 validators contained in the dUNL as of October 4, 2021.

Ripple used to control a larger fraction of validators listed in the dUNL. I give a historical overview of this fraction at the end of Section 5.2. Throughout a large majority of the history of the XRP Ledger, Ripple controlled more than 20% of validators in the dUNL. Moreover, its level of control was actually at 100% of validators in the dUNL for much of its history. This is relevant because, as discussed below in more details, when an organization controls more than 20% validators in the dUNL, it becomes a single point of failure and needs to be trusted by other organizations that use the same dUNL.

Here, it is important to repeat and emphasize the result of my analysis related to Question E1. Even though Ripple today controls less than 20% of validators, it is still a single point of failure that needs to be trusted by all participants who use the only dUNL to which the rippled v1.7.3 software defaults, and which is controlled by Ripple and disseminated at `http://vl.ripple.com`.

Finally, in answering Question E3 (see Sec. 5.3), I consider the risks that might arise in the hypothetical case of Ripple's disappearance and the effects it might have on the XRP Ledger.

If Ripple disappears, it may be impossible to continue securely publishing the dUNL on the web address that Ripple currently controls (`http://vl.ripple.com`). For example, if the registration of the `ripple.com` domain expires, the attacker could register the domain on the attacker's name, take over control of the

domain and publish non-intersecting dUNLs hence subverting key properties of the system. If participants decide to ignore the dUNL to avoid such an attack, they would need to make changes to the XRP Ledger consensus software, or consent on UNLs through human agreement.

Finally, in the case where Ripple disappears but the dUNL somehow continues to be published correctly at `http://vl.ripple.com`, there are still potential risks. Namely, even assuming the complete absence of malicious attacks, the correct functioning of the XRP Ledger as a system requires 80% of validators within the dUNL to operate correctly and without faults or disappearance from the system. With 41 validators in the dUNL, this means that the XRP Ledger will halt if 9 or more validators (i.e., over 20% of 41 validators) stop functioning. With Ripple controlling 6 out of these 41, it may seem that the XRP Ledger might continue to operate even without Ripple.

However, if Ripple disappears, other validators may disappear as well. For example, 9 universities which have received funding from Ripple under the umbrella of the University Blockchain Research Initiative (`https://ubri.ripple.com/`) operate validators listed in the dUNL. If Ripple disappeared, the funding would eventually stop too, and the universities may realistically stop operating validators, in particular since the XRP Ledger offers no In-Protocol Incentives.[3] Disappearance of only 3 out of these 9 validators operated by universities, combined with the disappearance of 6 validators operated by Ripple would be sufficient for the XRP Ledger network to halt. In addition to the 9 universities, at least 4 companies that received funding from Ripple also operate validators listed in the dUNL. Operation of these validators could also be compromised if Ripple disappears.

3 Background

In this section, we[4] describe the methodology for evaluating the decentralization of a given blockchain system (Section 3.1) and the necessary technical background behind the Bitcoin (Section 3.2) and Ethereum (Section 3.3) blockchains. This background is needed in order to answer question E1 as stipulated in the "Assignment" section (Section 1.1).

3.1 Methodology for Evaluating Decentralization in Distributed Systems

Decentralized *blockchain* systems are a subset (i.e., a special case) of *decentralized* systems, which are in turn a subset of *distributed* systems.

In computer science literature, a *distributed system* is loosely defined as *a collection of independent computers that appear to its users as a single coherent system [20].*

In turn, *decentralized systems* can be defined as *a subset of distributed systems where multiple authorities control different components and no authority is fully trusted by all [21].*

For instance, popular cloud and social networks like Google, Facebook or Twitter, are examples of distributed systems. However, these systems are not decentralized, as each of them is controlled by a single authority (company). Note that it is not sufficient for a system to simply have its components controlled by

[3] As discussed in Section 5.3, this argument could be extended to commercial companies, business partners of Ripple, which operate validators listed in the dUNL.

[4] Conforming to the style of scientific writing I have been used to, I sometimes use "we" instead of "I".

multiple authorities, to be classified as decentralized — the absence of a single trusted authority is needed, meaning that any component in a decentralized system could be *Byzantine*.

Byzantine [13] here refers to the ability of a participant or a component in a distributed system to deviate from the algorithm prescribed to them. This includes any behavior, including acting to purposefully attempt to disrupt the functioning of the system (in this case we talk about *attacks*). Byzantine behavior in literature is also sometimes also called, e.g., *adversarial*, *malicious*, or *arbitrary*. In this report, we sometimes use these notions for better readability. Moreover, we use the notion of *adversary*, to denote an authority, or group of authorities, that can orchestrate behavior of individual Byzantine components to mount attacks on the system.

As we will argue later in detail, one example of a decentralized system is the Bitcoin blockchain, in which no single authority, even if Byzantine, can subvert the correct functioning of the system.

Beyond the above basic definition of a decentralized system, computer science literature considers multiple *aspects* of decentralization in an attempt to refine and characterize its nuances, as well as the differences among decentralized systems (see e.g., [17] for a recent survey). We summarize these into the following *decentralization aspects* which we will later use to evaluate the decentralization of the Bitcoin blockchain, the Ethereum blockchain and the XRP Ledger.

1. **Resilience** of a system refers to its ability to withstand Byzantine behavior of components of the system.

   Resilience itself may apply to different properties of the system, namely *safety* and *liveness* [12, 1].

   Informally, a safety property of a system stipulates that "bad things" do not happen. An example of such a safety property in the context of blockchains is *double-spend* resistance [16] which, in short, requires the system to prevent an adversary from spending the same amount of money twice.

   In turn, a liveness property stipulates that "good things" do eventually happen. An important liveness property of a blockchain system is *censorship* resistance [9] which, in short, requires the system to prevent the adversary from excluding (censoring) payment transactions. Another important liveness property of a system is not to stop making progress in its operation altogether. For instance, if a blockchain halts and stops processing transactions, it fails to satisfy liveness.

   We define the censorship and double-spend resistance properties more precisely later, in Section 3.2.

   In this context, the scientific literature and engineering practice is typically interested in the minimum number of authorities that the adversary needs to compromise to subvert a key property of the system, such as safety or liveness. In the context of blockchains this number is sometimes referred to as the *Nakamoto coefficient*[5] [19, 23]. Intuitively, the higher the Nakamoto coefficient, the higher the level of decentralization. As per the definition of a decentralized system we adopted [21], if this number is 1 — i.e., if a single participating authority can compromise a key property of the system — the system cannot be deemed decentralized.

2. **Inclusiveness** of the system refers to the ability of the system to welcome new participants in a way which provides them with equal opportunities compared to existing participants [22]. In short, a decentralized system provides *Equal Opportunities* if it [22]:

[5] Honoring Bitcoin's pseudonymous inventor, Satoshi Nakamoto. Citation [23] is an example of a scientific paper that explicitly mentions the Nakamoto coefficient.

(a) allows any participant Alice to have an equal role in the system as any other (new or existing) participant Bob, provided Alice makes the same investment in system resources as Bob, and

(b) the system does not prevent Alice from making such an investment.

Then, a decentralized system is defined as Inclusive if and only if it satisfies Equal Opportunities [22].

Inclusiveness is a refinement of a well-known classification of blockchain systems into *permissioned* and *permissionless* systems (see e.g., [15]). In short, in permissionless systems, participants self-elect into the system, whereas permissioned systems rely on an external selection process to be admitted into the system, where *authority to choose [participants] typically resides with an institutional or organizational process [15].* In other words, permissionless systems are *open membership* systems, whereas permissioned systems are *closed membership* systems. Therefore, as a general principle, permissionless systems are to be considered more decentralized than permissioned systems. Moreover, permissioned systems are never Inclusive, while permissionless systems may or may not be Inclusive.

For example, some permissionless systems, including the XRP Ledger, allow anyone to participate but in a way that prefers some participants over the others. This makes them permissionless but not inclusive. In the XRP Ledger, nodes that participate in the system but which are included into the dUNL have a different role than the nodes which may elect to participate in the system but are excluded from the dUNL, violating Equal Opportunities.

Related to Inclusiveness, there are other approaches to refining the notion of permissionless systems in the scientific literature, which aim to capture the equality of participants within the system, taking into account the size of their investment. For instance, Karakostas et al. [10] define *egalitarianism* in a rather technically involved way aiming at capturing the proportionality of rewards of participants in blockchains compared to their investment. In a related approach, Fanti et al. [7] define *equitability*, which quantifies how much a participant can amplify her token holdings compared to her initial investment. As both notions of equitability and egalitarianism are based on participants' rewards, i.e., In-Protocol Incentives, they cannot be applied to the XRP Ledger, as the XRP Ledger does not have any rewards for participants in the system, unlike the Bitcoin and Ethereum blockchains.

Finally, some authors recognize *operational decentralization* as an important aspect [17] that is related to Inclusiveness. Intuitively, operational decentralization aims at capturing special hardware requirements for participation in the system — the less specialized the hardware requirements, the higher the decentralization. For instance, a system which requires large amounts of storage (e.g., hard disk space) to participate in blockchain A would be deemed more centralized than blockchain B which requires less storage space [17].

3. **In-protocol Incentives** is the decentralization aspect which refers to whether the system has rewards for protocol participants, paid out to protocol participants within the protocol itself. Such payments are typically in the protocol's *native token*, e.g., "BTC" on the Bitcoin blokchain. In-protocol incentives are an important aspect of decentralized systems [17]. Troncoso et al. [21] argue that the development of adequate incentives is necessary to build a successful decentralized system.

   In general, In-protocol Incentives test if the system is genuinely open to new participants. On the one hand, a permissionless system that provides incentives for participants will attract new participants,

particularly if it is Inclusive.

On the other hand, a permissionless system that does not provide In-Protocol Incentives is only seemingly open, as new participants have less or no economic rationale to join the system. Such a system may resort to out-of-protocol incentives, in which case incentives are not governed by system software, but typically by people. Out-of-protocol incentives may involve existing participants establishing business and contractual relations with new participants to motivate them to join the system. This approach resembles and is more common in permissioned networks [2].

In the context of incentives, wealth distribution across token stakeholders is also considered an aspect of decentralization [17]. If the tokens of a system are held widely among many holders, the system is more likely to be considered more decentralized. If there is concentration of ownership, the system is more likely to be considered more centralized.

4. **Governance** of the system refers to the level of power, if any, of human stakeholders to influence and change key rules in the system, e.g., through software updates.

   Several parameters for evaluating decentralization of governance power have been proposed or discussed in the literature. These include:

   (a) *governance of the infrastructure* [8], or *improvement control* [17], often involving the number of developers contributing to a system codebase and the number of people contributing to the discussion around a system's design [3],

   (b) *existence of a public face* [8], which can be defined as a personality and/or institution that is widely recognized as a spokesperson or a representative of the system.

   (c) *owner control*, measured by examining the total tokens accumulated by the stakeholders in the early adoption period [17].

Finally, some authors [17] consider additional aspects of decentralization, including the decentralization at the *network layer*, i.e., pertaining to the decentralization of the network that underlies a distributed system, and the decentralization at the *application layer*, which includes, e.g., the diversity of wallets and applications that permit users to interface with the assets on the blockchain. Decentralization at the network layer requires that no single authority can control all the participants of a decentralized system at the network and infrastructure layers. For instance, a system which is controlled (administered) by multiple organizations that all host their participating nodes on a single cloud provider (e.g., Amazon Web Services) is not to be considered decentralized, as the cloud provider itself could be seen as a single trusted authority.

To maintain emphasis on the core distributed systems aspects, in this report we acknowledge these decentralization aspects that go beyond the core of a system, namely network and application layer decentralization, yet we opt to focus on decentralization aspects of systems proper.

3.2 Bitcoin Blockchain

Bitcoin is an open-source peer-to-peer computer network (also known as the "blockchain") for generating and transferring (transacting) electronic coins (denoted by BTC) among users of the blockchain. BTC is the *native coin* of the Bitcoin blockchain — this means that BTC does not represent any concept outside

the Bitcoin blockchain and that participants in the system are rewarded only in BTC. In the following, we denote by "Bitcoin" the Bitcoin blockchain, i.e., the peer-to-peer computer network and its software, and by "bitcoin", or "BTC", its native electronic coin.

Bitcoin was conceived [16] as an electronic cash network to allow online payments to be sent directly from one party to another without going through a financial institution or any other trusted middleman. This was not possible prior to Bitcoin as all electronic payments required trusted intermediaries, unlike physical, in-person, cash or barter transactions. Namely, prior to Bitcoin, electronic payments over the internet were sent only using trusted intermediaries such as PayPal, credit card processor companies (e.g., AMEX, VISA, MasterCard) or through traditional banking payment systems in which banks act as trusted payment intermediaries.

At a high-level, in Bitcoin, a user Alice wishing to send 1 BTC to another user Bob, uses her private cryptographic key to digitally sign a transaction to transfer 1 BTC from an *address A*, that Alice controls, to *address B* supplied to Alice by user Bob. Alice's private cryptographic key is like a very long password known only to Alice, which is cryptographically tied to *address A*.

Knowledge of the private key allows Alice to have control over address A and over the BTC digitally represented at that address. As a fundamental principle, whoever controls the private keys corresponding to a given address, controls bitcoin pertaining to that address.

The main challenge in such a system arises when users are not trusted by other users. This lack of trust is inherent to a system without trusted intermediaries. Namely, Alice could attempt to *double-spend* her BTC.

Consider the following example of a double-spend attempt. Alice signs transaction $tx_{Alice-to-Bob}$ in which she transfers 1 BTC from address A she controls, to Bob's address B. However, she also signs a conflicting transaction $tx_{Alice-to-Alice}$ in which she sends 1 BTC from address A to another address A' that Alice also controls.

Which of these conflicting transactions should be actually taken into account is the main technical problem Bitcoin solves. In the process called *consensus*, peers in the Bitcoin network, without trusting each other, agree on the global order of all transactions in the system thanks to a set of predetermined parameters (programmed into the software that created the Bitcoin network) that govern how to reach consensus.

In our example, all peers in the Bitcoin network would agree on the relative order between the two conflicting transactions $tx_{Alice-to-Bob}$ and $tx_{Alice-to-Alice}$. The first transaction in that order would be considered valid, whereas the other would be discarded. Or, the order could be the other way around — the point is that the consensus mechanism for recording transactions on the Bitcoin blockchain (explained in detail later) provides a mechanism for participants in the network, who may not even know each other and do not trust each other, to nevertheless agree to validate the exact same sequence of transactions.

Besides preventing double-spends, another important property Bitcoin provides is censorship-resistance. In short, censorship-resistance guarantees a correctly-behaving user Alice to have her transactions eventually included in the blockchain (while possibly having Alice pay a *transaction fee* for this service). In other words, censorship-resistance guarantees that transactions will not be excluded from the Bitcoin blockchain due to actions of a Byzantine adversary or due to peers disappearing from the system.

In the following, we explain the Bitcoin consensus mechanism, first describing consensus preliminaries (Section 3.2.1) followed by explaining its validation mechanism (Sec. 3.2.2).

3.2.1 Bitcoin Blockchain Consensus — Preliminaries

For efficiency reasons, Bitcoin processes transactions in blocks, which are groups of transactions together with protocol metadata. Blocks have a maximum block size. Effectively, the Bitcoin consensus mechanism establishes a global order on those blocks forming a *chain* of blocks (i.e., a "blockchain"). Consequently, Bitcoin establishes global order on the transactions contained in those blocks.

Bitcoin software defines a so-called *genesis* block, the first block in the chain, to which the latter blocks are appended. Bitcoin genesis block contains a link to the "real" (physical) world, with the headline of the cover page of *The Times* (British daily national newspaper) from January 3rd, 2009 reading *"Chancellor on Brink of Second Bailout for Banks"* being written into the Bitcoin genesis block. This link to the real world, beyond possibly conveying a motivation for the existence of Bitcoin, is important because it proves that the creator of the Bitcoin network, Satoshi Nakamoto, could not have run the code before that day to generate blocks which would be considered valid by the Bitcoin blockchain.

At the beginning of the Bitcoin blockchain's history there were really no bitcoin to transact, as none had been brought to existence (i.e, *minted* or *mined*) yet. To bring bitcoin into existence, Bitcoin software defines a *block reward*, which is at the same time an incentive for participants to participate in Bitcoin consensus. Bitcoin rewards every participant who successfully adds a block to the blockchain with a fixed reward, which halves every 210,000 blocks. The period of 210,000 blocks corresponds roughly to 4 years, as Bitcoin block production time is set to self-adjust to an expected 10 minutes between consecutive blocks. For the first 210,000 blocks, the block reward was 50 BTC per block. With maximum bitcoin supply, as stipulated by Bitcoin code, being 21 million BTC, 50% of all bitcoin have been mined in the first 210,000 blocks.[6] With block reward halving to 25 BTC, from block 210,001 to block 420,000, an additional 25% of bitcoin total supply have been minted in that period, and so on, with the current Bitcoin block reward conveniently conveying which percentage of the total supply has been minted within the current 4-year window. Currently, more than 12 years after the genesis block, the Bitcoin network has produced over 700,000 blocks with the current block reward being 6.25 BTC.[7]

Once a block reward brings bitcoin into existence, bitcoin can be transacted. For instance, assume Alice won the block reward at block number 100,000. Then, starting from the next block 100,001, Alice can transact those bitcoin and send them to other participants.

A participant in the Bitcoin network is an entity that runs a *full node*. Such a participant is sometimes also called a *peer* or a *validator*. Each Bitcoin full node keeps the entire history of the blockchain, validates new blocks and (optionally) participates in creating new blocks. Bitcoin's maximum block size and a relatively conservative time period interval of 10 minutes between the blocks imply that the blockchain does not grow too fast compared to advances in computer hardware.

Today, the size of the Bitcoin blockchain is about 400 GB of data,[8] which means that a full node can be easily run on low-cost hardware, with a mid-sized hard-disk and internet connection, basically by anyone.[9] Moreover, users can entirely opt-out from running full nodes, by maintaining only *client* wallets,

[6] See, for example, an illustration on `https://static.coindesk.com/wp-content/uploads/2020/03/bitcoin-supply-and-subsidy-775x500.png`.

[7] The reward may be fractional, as each bitcoin is divisible into 100 million smaller units, usually called satoshis. As an illustration of the value of Bitcoin block reward incentives, awarded on average every 10 minutes, the market price of the 6.25 BTC block reward today is, roughly, about $300,000 USD.

[8] `https://blockchair.com/bitcoin/charts/blockchain-size`.

[9] Bitcoin full node can be run on hardware which today costs about $200 USD, see `https://getumbrel.com`.

which protect their private keys and send Bitcoin transactions to others' (full) nodes. Finally, full nodes are incentivized to invest more into hardware and computing equipment, if they wish to have a higher probability of obtaining block rewards in the context of Bitcoin consensus, as explained next.

3.2.2 Bitcoin Consensus Validation

Bitcoin consensus proceeds as follows [16]:

1. New proposed transactions are broadcast to all nodes.

2. Each node collects new transactions into a block. A node cryptographically links the new block to its predecessor (parent) block. These parent links define the position of the new block in the blockchain and its path all the way to the genesis block. In short, a node chooses the predecessor block for the new block to be the one which has the *longest chain*[10] of blocks on its path to the genesis block, out of all blocks known to a node. In principle, a Bitcoin node only considers as valid only those transactions contained in the longest chain.[11]

3. In the process often called *mining*, or *Proof-of-Work* [16], each node repeatedly tries to find a final piece of information, called a *nonce*, which when embedded into the new block, will make other nodes accept and declare the new block as *valid*.

   This is the **key point** in the otherwise relatively straightforward Bitcoin consensus. This part of Bitcoin consensus relies on the widely-established cryptographic primitive called *cryptographic hash function*, or simply a *hash function*. A hash function $H()$ is a deterministic function which takes as input data of any length, e.g., a Bitcoin block, or a picture of a cat, or a YouTube video, and outputs a fixed length string of bytes, which uniquely represents the original input data. A cryptographic hash function has a few "magical" properties which Bitcoin makes use of, in particular that one cannot predict the output of a hash function by changing slightly the input, nor can it construct the otherwise unknown input which gives the desired output.

   So how does the hash function help establish block validity?

   The Bitcoin consensus validation mechanism requires a hash of a valid block to start with a specific number of zeros (0s) when represented as a bit string, that is a sequence of 0s and 1s. However, since the output of a hash function cannot effectively be predicted, a block hash with one specific nonce appears basically as a random string of 0s and 1s. Therefore, nodes need to try many nonces in order to be lucky and construct the required final data for the block such that the hash of the block will start with many 0s, as required by the validation code.

   The actual required number of leading zeros is self-adjusted by the Bitcoin blockchain during its lifetime, based on the Bitcoin code and the frequency of mined blocks, to maintain an expected block time of 10 minutes between the blocks.

   In summary, finding a nonce which makes the block valid is effectively a very simple but computationally intensive guessing game in which a node repeatedly tries different nonces, applies them to the rest of

[10] In fact, it is the chain which requires most work, which is most often the longest chain. For simplicity of narrative, we talk about "longest chain."

[11] Some blocks may potentially end up on branches off the longest chain. These blocks are called *orphaned* and transactions in such blocks are invalid and not taken into account.

the block, applies the hash function and sees if the output hash has the required number of leading zeros.

4. When a node finds a nonce and completes the Proof-of-Work, it broadcasts the block to all other nodes.

5. Other nodes run the *validation step* and accept the block only if: *(i)* all transactions in it are valid and do not contain already spent bitcoin, and *(ii)* the hash of the block starts with the required number of 0s.

   Unlike the mining step (Step 3) which is computationally very expensive to compute, and is typically completed only by nodes with high computing power, this validation step (Step 5) is very simple and inexpensive to compute even on low-cost hardware.

To summarize, Bitcoin Proof-of-Work (Step 3 above) consists of a miner node performing repeatedly the following substeps: a) changing the nonce, b) applying the hash function, c) seeing if the output starts with the required number of 0s, and going back to substep a) if it does not. In recent months, the Bitcoin network as a whole is estimated to have performed anywhere between 68 EH/s (exahashes per second) on June 28, 2021 and 190 EH/s (on May 9, 2021).[12] An exahash per second is one quintillion (a billion billion) hashes per second, a very large number of operations.

3.2.3 Evaluating Bitcoin Decentralization

In this section we evaluate Bitcoin consensus as described in the previous section, in the context of the decentralization methodology introduced earlier in Section 3.1. This will help us answer question E1 for expert opinion as stated in Section 1.1.

Resilience. As discussed in Section 3.1, Resilience of a decentralized system can be measured with respect to different properties.

We look at two major possible issues: the double-spending issue and the censorship of transactions issue.

To mount these attacks effectively on the Bitcoin network, the adversary needs to control more than 50% of the network computing power. This would allow the adversary to simply ignore blocks produced by the rest of the network and produce the dominant longest chain, which would then, by Step 2 of the Bitcoin consensus protocol (Sec. 3.2.2), be the effective history of transactions. In the case of censorship attacks - this new history could simply be empty of transactions, or could specifically exclude the transactions of certain participants the adversary wishes to censor. This is known as a 51% attack for Bitcoin and requires a majority of the hash power of the network.

Whereas it is difficult to precisely calculate the Nakamoto coefficient (number of different authorities required to mount the attack) for Bitcoin, this resilience can be conservatively estimated. Namely, Bitcoin nodes often group into so-called *mining pools* to spread out their earnings from block rewards more evenly over time. While individual nodes are often not directly under the control of a mining pool operator authority and could leave the mining pool if they detected that they were participating in an attack, for a *very conservative* estimate of Resilience one can assume that a mining pool fully controls all the nodes inside the pool. With

[12] `https://www.coinwarz.com/mining/bitcoin/hashrate-chart`.

this in mind, at the time of writing this report, more than 50% of Bitcoin mining power is controlled by 4 mining pools.[13] Therefore, the conservative estimate of the Nakamoto coefficient for Bitcoin is 4.

Finally, it is worth noting, in the context of later comparison to the XRP Ledger and the impact of Ripple's hypothetical disappearance (Sec. 5.3), that in the absence of Byzantine participants, the Bitcoin network is resilient to any number of participants disappearing from the system. This was effectively tested in the Bitcoin network recently, when the computing power in the Bitcoin network dropped by about 65% between May 9, 2021 (190 EH/s) and June 28, 2021 (68 EH/s), as we already discussed. This had little effect on the Bitcoin network, except that, for some time between periodic network self-adjustments, block production took more than 10 minutes on average.

Inclusiveness. Bitcoin is a permissionless system which provides Equal Opportunities, because:

- Bitcoin allows any two participants, new or old, that make the same investment into system resources (computing power) to play the same role in the system.[14]
- Furthermore, the nature of Proof-of-Work consensus does not prevent any participant from making such an investment into system resources. In particular, assuming a free market for computing power, existing participants cannot prevent new participants from entering the system.

  With innovation in computing and the seemingly unstoppable growth of computing power available to humans, often modeled by Moore's Law (see e.g., [14]), the computing power of the existing participants actually decays in time compared to the computing power available outside the system, which is free to join the Bitcoin network.

Consequently, as it provides Equal Opportunities, Bitcoin is Inclusive.

Bitcoin also allows a large degree of operational decentralization, as its full node requirements are relatively modest with the only notable full node hardware requirement being a hard disk capable of storing 400 GBs of blockchain data for the full blockchain history (see also Sec. 3.2.1).

In-protocol Incentives. Bitcoin provides incentives to nodes to participate in the system. Besides block rewards which we discussed in Sec. 3.2.1, Bitcoin also awards block miners with *per-transaction fees*.

Incentives provide a rational and transparent economic reason for new participants to join a decentralized system. Combined with Inclusiveness, which means that the system welcomes new participants, such incentives contribute to the rise of new participants promoting decentralization.

Finally, as indicated in the Bitcoin whitepaper [16], the economic incentives of Bitcoin make safety attacks towards compromising Resilience less likely than if the In-Protocol Incentives did not exist. If certain nodes control a large amount of computing power in Bitcoin they have an economic dilemma between using that power to attack the system or using that power to behave correctly and earn block rewards and transaction fees. This intuitively contributes to increasing the Nakamoto coefficient (Resilience measure) and consequently increasing the decentralization level of the network, in the presence of economically rational participants.

[13] As we observed at `https://taproot.watch/miners` and `https://btc.com/stats/pool`.

[14] Note that participants that do not make the same investment into system resources, do not necessarily have the same power in the system. For instance those that invest more into computing power can expect higher rewards from the system (e.g., more frequent block rewards).

Governance. Concerning code improvement proposals, anyone can propose a change to the Bitcoin open-source software via Bitcoin Improvement Proposals (BIPs).[15] In practice, relatively few "core" developers (developers of the Bitcoin Core reference node software) propose and implement changes. Major changes to software are relatively rare, with no BIP containing a backwards incompatible change to Bitcoin consensus (also known as a hard-fork) ever having been deployed in the software. For changes that implement more strict consensus validation rules, i.e., which reduce the space of valid blocks and are backwards compatible (soft-fork), consensus among core developers is required, together with approval of miners through on-chain voting.

That said, as Bitcoin is open-source software, anyone can make any change to the software. A number of such backwards incompatible changes to Bitcoin code have resulted in Bitcoin network forks and, effectively, separate blockchain networks.[16]

The Bitcoin network does not have a single individual or company acting as its public face [8]. This fact contributes to its decentralization. The absence of a public face is primarily due to the fact that its creator(s) acted under the pseudonym *Satoshi Nakamoto*, who disappeared from the public discourse more than 10 years ago.

Regarding owner control, Bitcoin did not have a hidden owner accumulation phase. The first transaction in the Bitcoin network happened in block #170, seemingly between Satoshi Nakamoto and a cryptographer Hal Finney, on January 12, 2009, 9 days after The Times newspaper timestamp contained in the genesis block.[17] The first block following the genesis block was mined, probably by Satoshi Nakamoto, 6 days after the genesis block,[18] on January 9, 2009.[19]

3.3 Ethereum Blockchain

Ethereum was announced in a post on the online Bitcoin forum, *bitcointalk*, in early 2014 by Vitalik Buterin [4], with the post designating Buterin as the inventor of Ethereum. The post mentions the other 6 members of the original Ethereum team.

Compared to Bitcoin, the main novelty of Ethereum was the introduction of the capability to code more complex and more general applications on top of a decentralized consensus. As Buterin stated in the Ethereum announcement post [4]: *"Up until this point, the most innovation in advanced applications such as domain and identity registration, user-issued currencies, smart property, smart contracts, and decentralized exchange has been highly fragmented, and implementing any of these technologies has required creating an entire meta-protocol layer or even a specialized blockchain."* Ethereum provides a platform for the development of such applications, one on which different applications can co-exist. In the Ethereum parlance, these applications are called "smart-contracts."

In the same forum post, a pre-sale of Ethereum's native token, called ether or ETH, was announced.

[15] `https://github.com/bitcoin/bips`

[16] Examples include Bitcoin Cash and Bitcoin Gold.

[17] Sources that discuss this include `https://thehunt.btcorigins.com/moments/the-first-transaction/` and `https://themoneymongers.com/first-bitcoin-transaction/`. I verified myself, by examining the Bitcoin transaction history, that the first transaction between two addresses indeed happened in block #170, see `https://www.blockchain.com/btc/block/170`.

[18] `https://www.blockchain.com/btc/block/1`.

[19] As it is widely believed, Satoshi Nakamoto may have mined a sizeable number of bitcoin in the early days of the network following the genesis, as an early participant. The exact number is practically impossible to support with hard evidence. However, we do have hard evidence, in the very Bitcoin transaction history, that an overwhelming majority of those early bitcoin that could be attributed to Satoshi Nakamoto were never transacted on the network.

The Ethereum genesis block defined roughly 72 million ETH (see `https://etherscan.io/stat/supply`), out of which about 60 million ETH tokens were sold in a crowdsale process called an initial coin offering (ICO) which ran in the summer of 2014. In the Ethereum ICO, people transferred their bitcoin (31,529 BTC in total, see e.g., `https://icoprice.com/ethereum/`) to the Bitcoin network address controlled by the Ethereum team and were allocated in return roughly 60 million ETH in the Ethereum genesis block, which appeared about a year later, in late July 2015. The difference of 12 million ETH was allocated in the genesis block for funding further development of the network.

Within the network, the native token ETH on the Ethereum network is used to pay for the computation performed by the applications (smart contracts) that run on top of the Ethereum network. This is called "gas." The Ethereum network does not have a hard cap on ETH supply.

3.3.1 Ethereum Consensus and its Decentralization

Since its inception, Ethereum has been using a variant of Bitcoin's Proof-of-Work for consensus. The two consensus protocols differ in subtle technical details, notably with respect to the approach of rewarding miners who mine blocks which do not end up on the "longest chain." Besides this difference, Ethereum uses a shorter time interval between blocks (about 15 seconds). At a high-level, the two consensus protocols can be considered very similar.

That said, practically since its inception, Ethereum has been planning to switch to an alternative consensus model called Proof-of-Stake, with the first software updates to the Ethereum network in this direction taking place recently. As the decentralization level of a distributed system fundamentally depends on its underlying consensus protocol, we evaluate the decentralization of the Ethereum network assuming its current consensus protocol, i.e., the one based on Proof-of-Work. After this, we briefly reflect on the potential impact of a Proof-of-Stake consensus to Ethereum decentralization.

Resilience. With Proof-of-Work as its underlying consensus mechanism, the reasoning about Ethereum Resilience shares similarities to that of Bitcoin. At the time of writing of this report, more than 50% of Ethereum mining power is controlled by 3 mining pools, making the conservative estimate of the Nakamoto coefficient for Ethereum equal to 3.[20]

Inclusiveness. With Proof-of-Work as the underlying consensus, Ethereum is a permissionless system which satisfies Equal Opportunities, which makes it Inclusive.

When it comes to operational decentralization, storing the full history of the entire state on Ethereum network has relatively high storage requirements of over 5 TB for an *archive* node which cannot be run on current commodity (i.e., widely available) hardware. However, the Ethereum network allows the pruning of old states with nodes maintaining the current state of the network (*full nodes*) requiring less than 1 TB of storage, which is still amenable to commodity hardware.[21]

In-protocol Incentives. Ethereum provides block rewards to Proof-of-Work miners similarly to Bitcoin. It also provides rewards to miners who mine blocks which do not end up on the longest chain.[22] It also

[20]`https://etherscan.io/stat/miner?range=1&blocktype=blocks` and `https://etherchain.org/miner`.

[21]`https://ethereum.org/sk/developers/docs/nodes-and-clients/#recommended-specifications`

[22]These are so-called "uncle" blocks, which include some of the blocks which Bitcoin would considered as "orphaned."

incentivizes miners by awarding them per-transaction fees. These incentives provide a rationale for new participants to join the network and contribute to decentralization.

Governance. Different research papers have analyzed the process of Ethereum improvement proposals (EIPs) and compared it to that of Bitcoin [3, 17]. The two communities are in this sense largely similar, with decentralization measures somewhat in favor of Bitcoin [3, 17].

Ethereum routinely deploys backwards incompatible updates (hard-forks). One of them was a reaction to a hacker exploit which affected several millions of ETH in June 2016, changing network rules to effectively refund the affected tokens.[23] This aspect of Ethereum governance remains controversial and has led to an alternative blockchain network (an Ethereum network fork) in which this refund did not take place.[24]

Other notable differences of Ethereum with respect to Bitcoin pertaining to the Governance aspect are the following: 1) several reputable sources (e.g., [11] and [6]) consider the inventor of Ethereum, Vitalik Buterin, to be its public face and 2) Ethereum development was funded using the proceeds of the ICO. Furthermore, the initial token distribution (owner control) of Ethereum is considerably different from that of Bitcoin, with 72 million ETH being pre-allocated in its genesis block (to crowdfunders and the development team), as we already discussed.

Impact of Proof-of-Stake on Decentralization. Proof-of-Stake and Proof-of-Work consensus protocols have fundamentally different implications on the decentralization of the network. In short, in Proof-of-Stake, “miners” do not expend electrical energy for mining but vote with their monetary power proportional to the size of their investments in the native token, i.e., ETH in this case. This implies considerably different economical dynamics compared to Proof-of-Work [7] and may outright lead to violation of Equal Opportunities and, consequently, Inclusiveness [22]. This may in turn lead to increased centralization of the network. Detailed analysis of the impact of Proof-of-Stake on decentralization seems, however, outside the scope of this report as that change has not yet occurred, and is available elsewhere [22]. In the context of this report, we evaluate the Ethereum network with its current consensus mechanism, i.e., Proof-of-Work.

4 XRP Ledger Description (Answer to Prefatory Question P2)

In this section, we describe the key technical aspects behind the XRP Ledger. In particular, we explain the concept of validation and consensus in the XRP Ledger and the concept of *Unique Node Lists* (UNL) in the XRP Ledger. We thereby answer Prefatory Question (P2), as stated in Section 1.1.

4.1 Validation, Consensus and Unique Node Lists (UNLs)

For clarity, in this section (Sec. 4.1), my personal comments and remarks are clearly marked as “(MV: ⟨text of a comment/remark⟩).” The rest of the description contained in this section is taken solely from the material which I consider endorsed by Ripple and/or its employees:

[23] See, e.g., [24], as well as `https://www.coindesk.com/understanding-dao-hack-journalists`, `https://eng.ambcrypto.com/ethereum-co-founder-vitalik-buterin-delves-into-infamous-dao-hack/`, or `https://www.gemini.com/cryptopedia/the-dao-hack-makerdao`.

[24] Ethereum Classic.

1. Brad Chase and Ethan MacBrough. "Analysis of the XRP Ledger Consensus Protocol", arXiv:1802.07242v1, 20 Feb 2018. [5].

   Chase and MacBrough are, respectively, current and former employees of Ripple.

2. Official XRP Ledger documentation, available at `https://xrpl.org/docs.html`.

3. Blockchain daemon implementing XRP Ledger in C++ (i.e., XRP Ledger, or rippled reference implementation), available at `https://github.com/ripple/rippled`, and in particular its latest release at the time of writing of this report, i.e., release 1.7.3 of 27 August 2021, as available at `https://github.com/ripple/rippled/tree/release`. We refer to this software as "rippled v1.7.3."

4. Original whitepaper by David Schwartz, Noah Youngs and Arthur Britto. "The Ripple Protocol Consensus Algorithm", available at `https://ripple.com/files/ripple_consensus_whitepaper.pdf` [18]. Since this document is marked as of "historical interest" only, this material is used only where explicitly designated and in the context which is still valid today.

4.1.1 Validators and UNLs

The XRP Ledger is a distributed blockchain system, with XRP as its native token. The XRP Ledger faces the same challenges as other digital assets in preventing double-spending and ensuring network-wide consensus [5].

XRP Ledger *nodes*, also called *rippled servers*, maintain (some amount of) a globally ordered history of *ledgers*, which in turn contain transactions. Each ledger is numbered with a *ledger index* and builds on a previous ledger whose index is one less, going all the way back to a starting point called the genesis ledger. (MV: A ledger can simply be viewed as a block. Basically, a "ledger" is to XRP Ledger what block is to Bitcoin.) Ledgers are cryptographically linked to their parent (predecessor) ledgers using a cryptographic hash function.[25] (MV: However, the number of leading zeros in a hash of a ledger is irrelevant, unlike in Bitcoin.)

XRP Ledger nodes can be configured in several modes and roles[26]. This includes the role of a *validator*, designating a rippled server which participates in the consensus protocol, called the XRP Ledger Consensus Protocol.

Each validator *Alice* in the XRP Ledger must have a *validator list*, or a *Unique Node List*, denoted by UNL_{Alice}. UNL_{Alice} represents the list of other validators *Alice* listens to as part of the XRP Ledger Consensus Protocol [5]. (MV: Messages sent to *Alice* by validators other than those in her UNL have no effect on the state of node *Alice* in the XRP Ledger Consensus Protocol and are effectively ignored by *Alice*.)

Each validator identifies itself with a unique cryptographic key pair that must be carefully managed. (MV: A validator is in fact identified by other validators by its public key part of the unique cryptographic key pair. A validator must keep the private part of its cryptographic key pair secret.)

The XRP Ledger reference implementation, rippled, provides a list of "curated default" [18] UNLs (dUNLs) to all validators (MV: containing public keys of a curated list of validators).

The only dUNL configured in rippled v1.7.3, in lines 55 and 56 of the file `https://github.com/ripple/rippled/blob/1.7.3/cfg/validators-example.txt`, is the one published at a *validator list site*

[25] See `https://xrpl.org/ledger-header.html`.

[26] See `https://xrpl.org/rippled-server-modes.html`.

located at `https://vl.ripple.com`. (MV: **This implies that the rippled software makes it such that a validator defaults to the dUNL that is controlled and published by Ripple Labs, Inc.** Other UNL publishers, including Coil, a company financially related to Ripple, are listed only as examples in the commented out section of the mentioned *validators-example.txt* configuration file, in lines 27-31. However, rippled v1.7.3 software defaults exclusively to the dUNL published by Ripple. In other words, when a new validator wishes to enter into the XRP Ledger, the rippled software it downloads defaults to installing a UNL list that was selected by Ripple.)

According to `https://github.com/ripple/rippled/blob/1.7.3/src/ripple/app/misc/ValidatorSite.h`, the software fetches the latest published recommended validator lists from the validator list site at *regular intervals*.

In addition to actually installing the default UNL list for new servers and making them periodically fetch the latest validator list, Ripple strongly recommends[27], for production servers, using the file `https://github.com/ripple/rippled/blob/1.7.3/cfg/validators-example.txt` for validator list sites (MV: i.e., the one which defaults solely to `https://vl.ripple.com`).

4.1.2 Consensus and Validation

The XRP Ledger Consensus Protocol is described as a Byzantine fault-tolerant (BFT) protocol, which "must operate in the presence of faulty or malicious participants [validators]." This can include "not responding to messages, sending incorrect messages, and even sending different messages to different parties" [5]. In general, the XRP Ledger Consensus Protocol aims to tolerate Byzantine validators, so long as they are no more than 20% of the total number of validators in any single UNL.

The goal of the XRP Ledger Consensus Protocol is to provide consensus properties across different validators. Roughly speaking, these properties are related to double-spending prevention and censorship resistance. Formally, safety properties relevant to the XRP Ledger Consensus Protocol are *Agreement* and *Linearizability* [5], which essentially mandate that correct validators fully validate transactions in the same global order (hence preventing double spending). Liveness, or *Censorship-Resistance* as stated in [5], mandates that if a correct client (i.e., user that might or might not run a validator) broadcasts a transaction to all validators, then all correct validators eventually fully validate that transaction.

The XRP Ledger Consensus Protocol starts with clients submitting proposed transactions to one or more validators in the network, who in turn broadcast the transaction to the rest of the network. The XRP Ledger Consensus Protocol consists of three primary steps [5]: *Deliberation*, *Validation* and *Preferred Branch*.[28]

In these steps, validators exchange messages with each other. As we already mentioned, in the XRP Ledger Consensus Protocol a validator takes into account only messages sent to it by validators in its UNL. If a validator is unable to receive messages from more than 80% of the validators in its UNL, the protocol eventually halts and is unable to guarantee liveness.

For two validators to agree on the same global order of transactions, their UNLs must intersect (or overlap). Chase and MacBrough provide, in Section 4 of [5], analysis of the required UNL intersection across different validators, in order to guarantee safety and liveness. The analysis in [5] shows that to ensure safety **the XRP Ledger Consensus Protocol requires the intersection between any 2 UNLs to be over**

[27] See `https://xrpl.org/run-rippled-as-a-validator.html`.

[28] These protocols steps are fairly involved and we describe them in detail in Appendix B.

60% (page 15, [5]). This is regardless of the underlying network behavior and assuming standard XRP Ledger Consensus Protocol assumptions that the potential adversary can control up to 20% of validators in the intersection of any two UNLs.

Further analysis done by Chase and MacBrough in [5], shows that, **under certain circumstances, a much higher intersection between any two UNLs is needed for the correct operation of the XRP Ledger Consensus Protocol**.

In particular, they show [5] that **if a communication network can be unreliable** (in short, network is *unreliable* if it can drop or delay messages sent between otherwise correctly functioning validators), **the XRP Ledger Consensus Protocol requires over 90% intersection between any two UNLs to provide safety** (see page 18, [5]) **and a 100% intersection across UNLs to provide liveness** (i.e., to guarantee censorship-resistance and that the network does not eventually halt) even if no validator is Byzantine (see Example 9, page 19, [5]).

We postpone the details of this argument, due to its technicalities, to Appendix B, where **we also extend the analysis of [5] to show that the XRP Ledger Consensus Protocol does not guarantee liveness even if the UNL overlap is 100%, in the case of an unreliable network with a single Byzantine validator**. The consideration of this argument is, however, optional and is not necessary for our expert opinion which is presented in the next section.

5 Expert Opinion

In this section I give my expert opinion, answering the "Questions for Expert Opinion" E1, E2 and E3, listed in Section 1.1.

5.1 Question E1: To what extent is XRP Ledger centralized or decentralized compared to Bitcoin and Ethereum?

To answer this question we first evaluate the decentralization of the XRP Ledger using the methodology of Section 3.1.

5.1.1 Evaluating Decentralization of the XRP Ledger

Resilience. The main attack vector through which a single party can violate key properties of the XRP Ledger is the following one:

If the publisher of a default UNL (dUNL) on `https://vl.ripple.com` is corrupted (Byzantine) it can serve a different UNL to different validators, without the necessary intersection among UNLs. Please refer to Section 4.1.2 for different intersection requirements which range between 60% and 100% intersection between any 2 UNLs, depending on the assumed underlying network conditions and the relevant XRP Ledger property (safety or liveness).

As a simple example, **a corrupted dUNL publisher may serve totally different UNLs (i.e., 0% intersection) to different validators, preventing the correct operation of XRP Ledger**.

For this reason, **the Nakamoto coefficient for the XRP Ledger is 1**. This implies that the XRP Ledger fails to satisfy the basic definition of a decentralized system as there is a single party which needs to

be fully trusted by all [21]. Therefore, in my opinion, the **XRP Ledger is centralized**.

In addition, even if the publisher of dUNL is correct and acts in a proper manner, as per our analysis of Appendix B, a single Byzantine member listed in the dUNL, combined with an unreliable network, can violate liveness of the XRP Ledger Consensus Protocol even when all other validators are correct and all use a dUNL with 100% overlap.

We again note that this last observation is not necessary for our opinion that the XRP Ledger is a centralized system. It simply strengthens the argument.

Inclusiveness. By allowing anyone to join the network as a validator, the XRP Ledger qualifies as a permissionless blockchain (in the sense that it allows anyone to participate).

However, the **XRP Ledger is not Inclusive** because it does not provide equal opportunities for validators to become listed in a dUNL.

Another way to look at this is that the very existence of a dUNL is a root cause of inequality in the system. If the system would not specify any dUNL, this inequality would disappear. This would however jeopardize Resilience further, as XRP Ledger safety and liveness with honest validators, critically depends on the large intersection across UNLs that validators use.

Being permissionless without satisfying Equal Opportunities does not make a system truly permissionless. The XRP Ledger is essentially an "open" system which anyone can join, but where a few participants hand-picked by Ripple have special status (which stems from their inclusion in a dUNL), and the other participants merely follow the commands of these special participants.

In-Protocol Incentives. The **XRP Ledger provides no In-Protocol Incentives** to participants, old or new.

Assuming economically rational participants, financial incentives for new participants to join the system may therefore come only externally to the system (out-of-protocol incentives), arguably through activities of entities that already have a financial interest in the system.

Business and financial relationships between Ripple and other participants that run XRP Ledger validators listed in the dUNL published by Ripple give reasonable evidence and examples of such out-of-protocol incentives.

For instance, 9 out of 41 validators in the dUNL that Ripple publishes belong to universities that are part of the University Blockchain Research Initiative (UBRI) (`https://ubri.ripple.com/`). The universities from UBRI that are on Ripple's dUNL are: IIT Bombay, Korea University, University of Nicosia, University College London, University of North Carolina, Australian National University, UC Berkeley, and University of Waterloo. Ripple has funded these universities through UBRI.

Additionally, 3 validators listed in the dUNL published by Ripple are operated by companies funded by Ripple or Ripple-affiliated entities as their main sources of funding according to Crunchbase, the leading data source for investments in the technology sector. These include Coil[29], XRPL Labs[30] and Towo labs[31], the latter two being funded by Xpring, a Ripple initiative that invests in projects related to the XRP Ledger.[32].

[29] `https://www.crunchbase.com/organization/coil-technologies/investor_financials`
[30] `https://www.crunchbase.com/organization/xrpl-labs/company_financials`
[31] `https://www.crunchbase.com/organization/towo-labs/company_financials`
[32] `https://www.crunchbase.com/organization/xpring`

In addition, one other company (Bitso) was funded by an investment round led by Ripple and had a Ripple senior executive as one of its board members.[33]

To summarize, unlike with the Bitcoin or Ethereum blockchains, which offer rewards in the form of digital tokens to those that engage in the blockchain validation process, the XRP Ledger provides no such incentives or rewards, which means that validators do not come organically to the XRP Ledger.

Governance. According to statistics available at `https://github.com/ripple/rippled/graphs/contributors`, the overwhelming majority of code commits and lines of code comes from the developers who are or have been affiliated with or funded by Ripple Labs, Inc. or predecessor companies.

XRP Ledger has a public face in Ripple Labs, Inc.

Regarding owner control (of initial tokens), the information is not available from the genesis ledger of the XRP Ledger as due to a bug ("mishap in the XRP Ledger history"[34]), ledgers 1 through 32569 were lost. According to the information about XRP Sales available at `https://xrpl.org/xrp.html`, "The XRP Ledger was built over 2011 – early 2012 by Jed McCaleb, Arthur Britto and David Schwartz. In September 2012, Jed and Arthur, along with Chris Larsen, formed Ripple (the company, called OpenCoin Inc. at the time) and decided to gift 80 billion XRP to Ripple in exchange for Ripple developing on the XRP Ledger." The maximum supply of XRP is 100 billion. The rest of 20 billion early XRP were, according to multiple public sources,[35] distributed among founders.

Therefore, we can conclude that 100% of the initial/total supply was under *owner control*, comprising Ripple Labs (i.e., its predecessor companies) and its founders. This clearly goes against decentralization, particularly when combined with absence of In-Protocol Incentives, as it limits the economic rationale for new participants to organically join the system.

5.1.2 Answer to Question E1: Comparison to Bitcoin and Ethereum

The XRP Ledger is centralized compared to Bitcoin and even Ethereum. Even if we evaluate the XRP Ledger outside the context of Bitcoin and Ethereum, it cannot be deemed decentralized and hence is centralized.

In short, unlike Bitcoin and Ethereum, the XRP Ledger is centralized as it takes corrupting only a single party to be able to compromise key properties of the system. Also, when considering the other decentralization aspects analyzed, the XRP Ledger evaluates worse and is more centralized than Bitcoin and Ethereum.

The summary of key decentralization aspects according to our analysis from Section 3.2.3 (Bitcoin), Section 3.3.1 (Ethereum) and Section 5.1.1 (XRP Ledger) is shown below, repeating for convenience Table 1 from Section 2.

[33] See `https://livenet.xrpl.org/validators/nHBidG3pZK11zQD6kpNDoAhDxH6WLGui6ZxSbUx7LSqLHsgzMPec` and `https://ripple.com/insights/our-investment-in-bitso/`.

[34] See https://xrpl.org/intro-to-consensus.html.

[35] See, for example, `https://blog.bitmex.com/the-ripple-story/`.

Decentralization aspect	**Ideal Decentralized System**	**Bitcoin Blockchain**	**Ethereum Blockchain (with Proof-of-Work)**	**XRP Ledger**
Nakamoto coefficient (Resilience)	always greater than 1, the higher the better	≥ 4	≥ 3	1
Inclusiveness	yes	yes	yes	no
In-Protocol Incentives	yes	yes	yes	no
Governance (public face)	no	no	yes	yes
Governance (tokens allocated at genesis)	0, the lower the better	0%	61.5% (about 10% owner controlled) of today's supply	100% (all owner controlled)

Table 1: Comparison of the XRP Ledger to the Bitcoin and Ethereum blockchains for key aspects of decentralization defined in the decentralization evaluation methodology of Section 3.1.

5.2 Question E2: To what extent have Ripple's efforts been needed to support the proper functioning of the XRP Ledger?

Ripple's effort needed *today* to support the proper functioning of the XRP Ledger, based on the current rippled software, includes:

1. Publishing a dUNL at `https://vl.ripple.com`. This includes maintaining security and ownership of ripple.com domain so an adversary cannot control a dUNL.

   This also includes making efforts to carefully change the published dUNL, even in the absence of actions of a malicious adversary. In a known incident that occurred in November 2018, which was also the topic of the May 26, 2021 deposition of David Schwartz in front of this court (pages 222-226 and Exhibit 44 therein), the XRP Ledger was stalled from making forward progress when one UNL expired and a new one was published.

   As indicated by an xrpchat online forum post which appeared later, in October 2020, from a user who appears to be Ripple's employee Nik Bougalis[36], following this November 2018 incident he *"personally restarted several validators,"* and *"the team at Ripple invested a significant amount of time troubleshooting the issue and proposed several improvements,"* illustrating the amount of human and in particular Ripple employees' effort needed to rectify the network halt in a case where changes in the published dUNL are not handled well.

2. Because of possible attacks on the network, that could result in safety or liveness violations, including the attack we describe in Appendix B, so long as it publishes a dUNL, Ripple needs to continue to curate and attest validators that it includes in a dUNL.

3. As of October 4, 2021, Ripple appears to directly control 6 out of 41 validators in the published dUNL. Due to possible Byzantine attacks, including the one we describe in Appendix B, Ripple needs to maintain security over these validators and ensure they behave honestly.

[36]See `https://www.xrpchat.com/topic/28872-the-network-is-down/?do=findComment&comment=850670`.

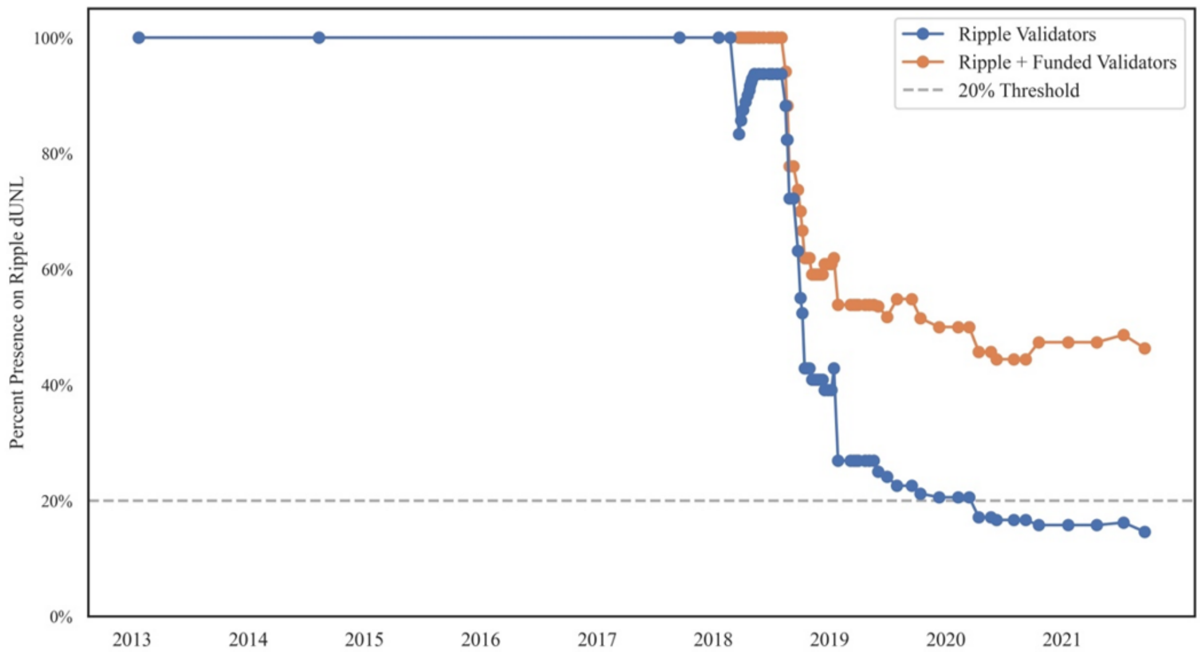

Figure 1: Ripple validators and validators operated by entities funded by Ripple, given as a fraction of the dUNL membership over time.

With modifications of software, points 1 and 2 above can, in principle, be done by an entity different from Ripple. Nevertheless, the XRP Ledger would still be centralized as this entity would still need to be fully trusted in the sense of the arguments pointed out in items 1 and 2.

In relation to point 3 above, it is worth noting that Ripple used to control a larger fraction of validators listed in the dUNL it publishes compared to the fraction it controls today. This fraction was even 100% for over half of the XRP Ledger history.

Figure 1 gives the change in time of the fraction of validators in the Ripple's dUNL belonging to Ripple as well as that of the fraction of validators belonging to Ripple or entities funded by Ripple.[37] These entities and their relation to Ripple are discussed in more detail in the next section, Section 5.3.

5.3 Question E3: What risks to the XRP Ledger would or might materialize if Ripple "walked away" or "disappeared"?

Like the previous question, we will answer this question assuming no software changes (i.e., assuming rippled v1.7.3). In short, if Ripple would disappear, serious risks related to the correct operation of the XRP Ledger network may arise.

We consider two cases: A) Ripple disappears and the network is still able to agree on the contents of the

[37] The main source for the data depicted in Figure 1 is obtained from `https://github.com/ripple/vl`, which contains validator public keys of every historical dUNL and the current dUNL. Validator ownership is classified using their respective domains found on `https://livenet.xrpl.org/validators/{publickeyofvalidator}`. Domains ownership was confirmed using the validator registry `https://xrpcharts.ripple.com/#/validators` and through `https://xrpscan.com/{public_key}`, or Google search of the public keys.

dUNL as currently published on `https://vl.ripple.com`, and B) Ripple disappears and leaves the network without a common UNL — that is, UNLs used by validators in the network change over time.

Consider the first case, case A:

- In the case where more than 20% of validators in the dUNL disappear, the network would not be operational. The current dUNL (as of October 4, 2021) contains 41 validators (data obtained from `https://xrpcharts.ripple.com/#/validators`).

  Hence, the network would cease to be operational if nine validators disappeared. Six validators are controlled by Ripple, i.e., they are shown to be resolving at a Ripple domain `validator.ripple.com`. In addition, many validators belong to entities which are funded by or have business relationships with Ripple, as we discussed in Section 5.1.1 (in the part regarding incentives).

  For instance, 9 out of 41 validators belong to universities part of the University Blockchain Research Initiative (`https://ubri.ripple.com/`). Ripple has funded these universities. If Ripple disappears, there is a risk that universities might cease to operate validators in absence of further funding. Three of such validators disappearing, in addition to Ripple's six, are sufficient for the network under the current dUNL to cease to be operational.

  Similar arguments can be made about the validators run by entities other than universities which have received significant funding from Ripple.

  For completeness, the list of validators controlled by Ripple and entities funded by Ripple, as well as the list of all 41 validators contained in the current dUNL are given in Appendix A.

- In addition, there is a separate risk that a validator in the common dUNL becomes compromised and Byzantine, enabling it to mount attacks against the network, such as the attack on liveness described in Appendix B.

  If Ripple is not there to evict such a validator from the dUNL, validators need to come up with different UNLs. This essentially reduces to the case B we consider next.

Consider now the second case, case B. In absence of the common UNL, network validators need to choose UNLs either by themselves, or based on some out-of-band communication with other validators.

If they choose UNLs themselves, they risk not getting a sufficient intersection among UNLs, jeopardizing the core properties of the XRP Ledger, safety and liveness. There is a high risk of state and ledger history forks in such a situation.

If they rely on out-of-band communication (i.e., outside the rippled software) with other validators and possibly entities external to the XRP Ledger to agree on a UNL, this could be done using software other than the XRP Ledger, or using human effort and communication. Using software other than XRP Ledger would basically imply another consensus (agreement) protocol, and could be viewed then as a change in XRP Ledger (rippled) software. The other option would be using human effort and communication to ensure agreement on sufficient intersection among UNLs (e.g., by relying on communication among human operators of individual validators). This defeats the very purpose for the existence of a software system that aims to implement distributed consensus.

6 Conclusions

In the context of the prefatory questions, I have been asked to explain the operation of consensus and validation in blockchain systems and, in particular, on the XRP Ledger. I have explained the concept of Proof-of-Work based consensus, used in Bitcoin, which is not based on validator identities, but rather relies on provable expenditure of a real-world resource (energy), and how it leads to a decentralized system.

In contrast, the consensus used in the XRP Ledger is based on a very different approach which puts validator identities at the core of the system.

In the case of the XRP Ledger this approach is technically executed in a manner contrary to decentralization principles, with a central authority controlled by Ripple given a task of publishing what can be seen as a special list of privileged validators.

With this in mind, it is easy to see that the answer to the expert opinion I was asked to provide—whether the XRP Ledger is a decentralized or centralized system—is that the XRP Ledger does not satisfy a basic definition of a decentralized system. To be decentralized, participants need not trust any single party. For the XRP Ledger, participants need to trust at least one other party, which is currently Ripple as the publisher of the dUNL to which the XRP Ledger software defaults.

To evaluate XRP Ledger characteristics related to decentralization in more depth, and to answer expert questions I have been asked to opine on, I surveyed scientific literature. The scientific treatment of the notion of decentralization has advanced in recent years to give a precise minimal definition of a decentralized system, as well as a more refined, general taxonomy of decentralized systems.

Summarizing this literature, I identified four decentralization aspects (Resilience, Inclusiveness, In-Protocol Incentives, and Governance) as, in my opinion, the most relevant ones. I based the methodology for evaluating the decentralization of distributed systems around those aspects, and I have evaluated Bitcoin, Ethereum and XRP Ledger through their lens.

XRP Ledger scores poorly in these aspects compared to Bitcoin and to Ethereum, which itself evaluates as more centralized than Bitcoin. The Resilience of the XRP Ledger is poor as it requires trusting a single party. It further is not Inclusive, as it makes distinctions among participants and does not provide them with equal opportunities. It has no In-Protocol Incentives, leaving the incentivization of new participants towards increasing the system size in the hands of entities that already have financial interest in the system, such as Ripple Labs Inc. Finally, its Governance related measures are poor.

In answering further questions for my expert opinion, I have identified the efforts required by Ripple towards the proper functioning of the XRP Ledger, as well as identified the risks that may arise in the case of Ripple's hypothetical disappearance. In short, in this case, serious risks related to the correct operation of the XRP Ledger network may arise.

The opinions expressed in this report are based on my review and analysis of the documents that I have reviewed. I reserve the right to supplement my report and analysis based on any new evidence brought to my attention.

[redacted]

References

[1] B. Alpern and F. B. Schneider. Recognizing safety and liveness. *Distributed Comput.*, 2(3):117–126, 1987.

[2] [redacted]

[3] S. Azouvi, M. Maller, and S. Meiklejohn. Egalitarian society or benevolent dictatorship: The state of cryptocurrency governance. In A. Zohar, I. Eyal, V. Teague, J. Clark, A. Bracciali, F. Pintore, and M. Sala, editors, *Financial Cryptography and Data Security - FC 2018 International Workshops, BITCOIN, VOTING, and WTSC, Nieuwpoort, Curaçao, March 2, 2018, Revised Selected Papers*, volume 10958 of *Lecture Notes in Computer Science*, pages 127–143. Springer, 2018.

[4] V. Buterin. [ANN] Ethereum: Welcome to the beginning. `https://bitcointalk.org/index.php?topic=428589.0`, 2014.

[5] B. Chase and E. MacBrough. Analysis of the XRP ledger consensus protocol. *CoRR*, abs/1802.07242, 2018.

[6] Encyclopaedia Britannica. 20 under 40: Young shapers of the future (business and entrepreneurship). `https://www.britannica.com/list/20-under-40-shapers-of-the-future-in-business-and-entrepreneurship`, 2021.

[7] G. C. Fanti, L. Kogan, S. Oh, K. Ruan, P. Viswanath, and G. Wang. Compounding of wealth in proof-of-stake cryptocurrencies. In I. Goldberg and T. Moore, editors, *Financial Cryptography and Data Security - 23rd International Conference, FC 2019, Frigate Bay, St. Kitts and Nevis, February 18-22, 2019, Revised Selected Papers*, volume 11598 of *Lecture Notes in Computer Science*, pages 42–61. Springer, 2019.

[8] P. D. Filippi and B. Loveluck. The invisible politics of bitcoin: Governance crisis of a decentralized infrastructure. *Internet Policy Review*, 5(4), 2016.

[9] A. E. Gencer, S. Basu, I. Eyal, R. van Renesse, and E. G. Sirer. Decentralization in bitcoin and ethereum networks. In S. Meiklejohn and K. Sako, editors, *Financial Cryptography and Data Security - 22nd International Conference, FC 2018, Nieuwpoort, Curaçao, February 26 - March 2, 2018, Revised Selected Papers*, volume 10957 of *Lecture Notes in Computer Science*, pages 439–457. Springer, 2018.

[10] D. Karakostas, A. Kiayias, C. Nasikas, and D. Zidros. Cryptocurrency egalitarianism:a quantitative approach. In *International Conference on Blockchain Economics, Security and Protocols (Tokenomics 2019)*, 2019.

[11] A. J. Kolber. Not-so-smart blockchain contracts and artificial responsibility. *Stanford Technology Law Review*, 198, 2018.

[12] L. Lamport. Proving the correctness of multiprocess programs. *IEEE Trans. Software Eng.*, 3(2):125–143, 1977.

[13] L. Lamport, R. E. Shostak, and M. C. Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982.

[14] C. E. Leiserson, N. C. Thompson, J. S. Emer, B. C. Kuszmaul, B. W. Lampson, D. Sanchez, and T. B. Schardl. There’s plenty of room at the top: What will drive computer performance after moore’s law? *Science*, 368(6495):eaam9744, 2020.

[15] A. Miller. *Permissioned and Permissionless Blockchains*, pages 193–204. 2019.

[16] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. `https://bitcoin.org/bitcoin.pdf`, 2008.

[17] A. R. Sai, J. Buckley, B. Fitzgerald, and A. LeGear. Taxonomy of centralization in public blockchain systems: A systematic literature review. *Information Processing and Management*, 58(4), July 2021.

[18] D. Schwartz, N. Youngs, and A. Britto. The ripple protocol consensus algorithm. `https://ripple.com/files/ripple_consensus_whitepaper.pdf`.

[19] B. Srinivasan. Quantifying decentralization. Blockstack Summit 2017, `https://www.youtube.com/watch?v=4UXT5YVJwB4`, 2017.

[20] A. S. Tanenbaum and M. van Steen. *Distributed systems - principles and paradigms, 2nd Edition.* Pearson Education, 2007.

[21] C. Troncoso, M. Isaakidis, G. Danezis, and H. Halpin. Systematizing decentralization and privacy: Lessons from 15 years of research and deployments. *Proc. Priv. Enhancing Technol.*, 2017(4):404–426, 2017.

[22] [redacted]

[23] Q. Lin, C. Li, X. Zhao, and X. Chen. Measuring decentralization in bitcoin and ethereum using multiple metrics and granularities. In *2021 IEEE 37th International Conference on Data Engineering Workshops (ICDEW)*, pages 80–87, 2021.

[24] C. L. Reyes, N. G. Packin, and B. Edwards. Distributed governance. *William & Mary Law Review Online*, 59(1). `https://scholarship.law.wm.edu/cgi/viewcontent.cgi?article=1003&context=wmlronline`.

[25] I. Amores-Sesar, C. Cachin, and J. Micic. Security analysis of ripple consensus. In Q. Bramas, R. Oshman, and P. Romano, editors, *24th International Conference on Principles of Distributed Systems, OPODIS 2020, December 14-16, 2020, Strasbourg, France (Virtual Conference)*, volume 184 of *LIPIcs*, pages 10:1–10:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

A Lists and Statistics of Validators Included in the dUNL published by Ripple, as of July 16, 2021

In this Appendix, we give lists and statistics related to validators included in the dUNL published by Ripple at `https://vl.ripple.com` (referred to as Ripple's dUNL), as of July 16, 2021 update.

Figure 2 gives the list of 19 validators belonging to entities funded by Ripple, whereas Figure 3 gives the list of all 41 validators.

Table: Validators Belonging to Entities Funded by Ripple

Entity	Domain	Connection to Ripple
Ripple	validator.ripple.com	Validator belongs to Ripple
Ripple	validator.ripple.com	Validator belongs to Ripple
Ripple	validator.ripple.com	Validator belongs to Ripple
Ripple	validator.ripple.com	Validator belongs to Ripple
Ripple	validator.ripple.com	Validator belongs to Ripple
Ripple	validator.ripple.com	Validator belongs to Ripple
Australian National University	xrp-col.anu.edu.au	Received funding through Ripple's University Blockchain Research Initiative[1]
IIT Bombay	isrdc.in	Received funding through Ripple's University Blockchain Research Initiative[1]
Korea University	blockchain.korea.ac.kr	Received funding through Ripple's University Blockchain Research Initiative[1]
UC Berkeley	shadow.haas.berkeley.edu	Received funding through Ripple's University Blockchain Research Initiative[1]
University College London	students.cs.ucl.ac.uk	Received funding through Ripple's University Blockchain Research Initiative[1]
University of Kansas	ripple.ittc.ku.edu	Received funding through Ripple's University Blockchain Research Initiative[1]
University of Nicosia	xrp.unic.ac.cy	Received funding through Ripple's University Blockchain Research Initiative[1]
University of North Carolina	ripple.kenan-flagler.unc.edu	Received funding through Ripple's University Blockchain Research Initiative[1]
University of Waterloo	ripplevalidator.uwaterloo.ca	Received funding through Ripple's University Blockchain Research Initiative[1]
Bitso	bitso.com	Ripple led an investment round and has a Senior Executive on Bitso's Board[2]
Coil	coil.com	Ripple (via Xpring) provided initial funding of 1 billion XRP ($265 million)[3]
Towo Labs	towolabs.com	Ripple (via Xpring) is lead investor[4]
XRPL Labs	validator.xrpl-labs.com	Ripple (via Xpring) is listed as sole investor[5]

[1] https://ubri.ripple.com/
[2] https://ripple.com/insights/our-investment-in-bitso/
[3] See https://www.crunchbase.com/organization/coil-technologies/investor_financials and https://cointelegraph.com/news/ripples-xpring-gives-265-mil-in-xrp-to-content-platform-coil
[4] See https://ripple.com/insights/investing-in-towo-labs/ and https://www.crunchbase.com/organization/towo-labs/company_financials
[5] See https://ripple.com/insights/doubling-down-on-xrpl-labs/ and https://www.crunchbase.com/organization/xrpl-labs/company_financials

Figure 2: The list of 19 validators listed in the Ripple's dUNL, belonging to Ripple or entities funded by Ripple.

Table: List of 41 Validators as of July 16, 2021 dUNL Update

Entity	Domain	Public Key
Alloy Networks	alloy.ee	[redacted]
AT TOKYO	www.attokyo.com	
Australian National Univ.	xrp-col.anu.edu.au	
Bahnhof	www.bahnhof.se	
Bithomp	bithomp.com	
Bitrue	www.bitrue.com	
Bitso	bitso.com	
Blockdaemon	arrington-xrp-capital.blockdaemon.com	
Cabbit Technology	cabbit.tech	
Eminence	verum.eminence.im	
Coil	coil.com	
Coinfield	xrp.coinfield.com	
Data443	data443.com	
Individual	digifin.uk	
Flagship Solutions Group	flagshipsolutionsgroup.com	
FTSO.eu	xrpvalidator.ftso.eu	
Gatehub	validator.gatehub.net	
IIT Bombay	isrdc.in	
Individual	jon-nilsen.no	
Kompany	brex.io	
Korea University	blockchain.korea.ac.kr	
NTT Data	ripple.ntt.com	
Peer Island	peerisland.com	
Ripple	validator.ripple.com	
Ripple	validator.ripple.com	
Ripple	validator.ripple.com	
Ripple	validator.ripple.com	
Ripple	validator.ripple.com	
Ripple	validator.ripple.com	
rippleitin	rippleitin.nz	
Telindus	ripple.telinduscloud.lu	
Towo Labs	towolabs.com	
UC Berkeley	shadow.haas.berkeley.edu	
University College London	students.cs.ucl.ac.uk	
University of Kansas	ripple.ittc.ku.edu	
University of Nicosia	xrp.unic.ac.cy	
University of North Carol.	ripple.kenan-flagler.unc.edu	
University of Waterloo	ripplevalidator.uwaterloo.ca	
Worldlink	validator1.worldlink-us.com	
XRP Scan	aloha.xrpscan.com	
XRPL Labs	validator.xrpl-labs.com	

Figure 3: The list of all 41 validators in the Ripple's dUNL.

B Details of the XRP Ledger Consensus Protocol, Including Vulnerability to Single Byzantine Validator with Completely (100%) Overlapping UNLs

In the rest of this appendix, we use the following definitions.

- A validator is called *correct*, if it operates without outages and follows the unmodified XRP Ledger Consensus Protocol protocol.
- A validator is called *Byzantine*, if its local copy of the XRP Ledger Consensus Protocol protocol is modified such that the validator deviates from the XRP Ledger Consensus Protocol protocol.
- The network is called *unreliable*, if it can drop or delay messages exchanged among correct validators.
- UNLs are said to *overlap completely*, or have *100% overlap*, if all UNLs of all correct validators are identical.

In the following, we provide details and in-depth analysis of the XRP Ledger Consensus Protocol. In particular, we:

1. Give the details behind XRP Ledger Consensus Protocol necessary for the in-depth analysis (Section B.1).
2. Summarize the analysis of liveness done by Chase and MacBrough in [5] (Section B.2).
3. Present our analysis, which shows that XRP Ledger Consensus Protocol fails to guarantee liveness, even with 100% overlap across all UNLs, if one validator in the said UNL can be Byzantine and if the network is unreliable (Section B.3).

B.1 Details of the XRP Ledger Consensus Protocol

The XRP Ledger Consensus Protocol consists of 3 main steps: *Deliberation*, *Validation* and *Preferred Branch* [5].

1. **Deliberation.** In this step, a validator *Alice iteratively* proposes a transaction set to include in the current ledger (i.e., block of transactions), based on transaction proposals received from other nodes in her UNL.

   When "enough" validators in validator's UNL propose the same transaction set, a validator generates the next ledger L, applies L to the current state, issues a *validation message* for L, exits deliberation, and proceeds to the Validation step.

   The notion of "enough" validators here depends on a particular subphase of the deliberation step and can be 50%, 65%, 70% or 95% of validators [5].

   The exact percentages mentioned above are to a large extent irrelevant as the correct execution of the protocol does not depend on the outcome of the deliberation step. Namely, as stated in the paper by Chase and MacBrough [5] on page 16: *"...deliberation can terminate with an arbitrary result.*

In practice, this may require a significantly degraded network, but is nonetheless a real risk. From a theoretical perspective, deliberation is therefore completely irrelevant; it is purely an optimization ... and it could be removed without fundamentally changing the protocol."

For illustration, Example 5 of [5] shows an example scenario where UNLs overlap completely (i.e., at 100%) and all validators are correct.In that example, due to an unreliable network, one group of validators can exit deliberation by validating ledger L and the other group of validators ledger L' different from L, at the same ledger index. We refer to this scenario, to which we will come back later, as *Network Split in Deliberation*.

In conclusion, under an unreliable network, at the end of the deliberation step, correct validators may well end up validating different ledgers and, in particular, end up in Network Split in Deliberation.

2. **Validation.** In this step a validator simply listens for validation messages coming from other validators from its local UNL. If a correct validator sees a *quorum* of validation messages for a ledger L, then it *fully validates* L.

   A quorum in XRP Ledger Consensus Protocol is defined as at least 80% of the nodes in a validator's UNL.[38]

   Once this happens, that ledger L and its ancestors are deemed fully validated and its state is authoritative and irrevocable.

3. **Preferred Branch.** In times of unreliable network or Byzantine failures of validators, it may happen that some correct validators fail to receive a quorum of validation messages for any individual ledger to fully validate.

   In short, a correct validator may see validation messages for two or more *conflicting ledgers*, which lie on different branches in the block history. In the case of conflicting ledgers, *Preferred Branch* is the step of the XRP Ledger Consensus Protocol which determines which of the ledgers and the corresponding branch of ledgers, the correct validator should switch to and consideras the "right" one.

   The details of the Preferred Branch are a fairly involved part of the XRP Ledger Consensus Protocol—we omit the details for the sake of clarity. What is important for the rest of this report is that a validator cannot switch the preferred branch from the one on which ledger L is, if that validator gets more than 50% of validations messages from the nodes in its UNL for some *descendant* of L [5].

   Here, a descendant of ledger L is recursively defined as: 1) either ledger L itself, or 2) another ledger which has L or some of L's descendants as a parent.

B.2 Liveness Analysis by Chase and MacBrough [5]

The analysis in Example 9 of [5], further shows that the XRP Ledger Consensus Protocol, under an unreliable network which causes Network Split in Deliberation, **fails to guarantee liveness (Censorship-Resistance) even with no Byzantine validators, unless the overlap of UNLs is 100%**.

Example 9 of [5] is illustrated below, in Figure 4, which is taken directly from [5].

This example considers:

[38] See `https://github.com/ripple/rippled/blob/release/src/ripple/consensus/ConsensusParms.h`, lines 73-74, in addition to [5].

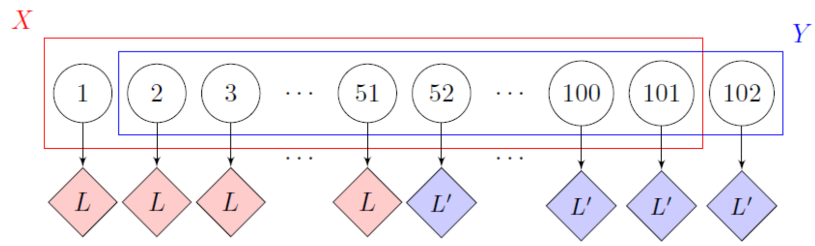

Figure 6: Example of stuck network with 99% UNL overlap and no Byzantine faults.

Example 9.

Consider a network of 102 peers drawin in figure 6. There are two UNLs, the red $X = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_{101}\}$ and blue $Y = \{\mathcal{P}_2, \mathcal{P}_3, \ldots, \mathcal{P}_{102}\}$. Peers $1-51$ use X and peers $52-102$ use Y. There are two ledgers, L and L'. The nodes listening to X all validate a descendant of L, while the nodes listening to Y all validate a descendant of L'. Since $51 > 0.5|X|$ nodes in X validate a descendant of L. Thus according to the preferred branch protocol all, the nodes listening to X cannot switch branch to L'. Similarly, since $51 > 0.5|Y|$ nodes in Y all validate a descendant of L, the nodes listening to Y cannot switch branch to L'. The network cannot ever rejoin without manual intervention.

Figure 4: Example 9 and Figure 6 from [5].

1. 102 validators experiencing Network Split in Deliberation;
2. Validators 1...51 use UNL X and send validation for descendant of ledger L;
3. Validators 51...102 use UNL Y and send validation for descendant of ledger L';
4. UNL X contains validators 1...101, in total 101 validators;
5. UNL Y contains validators 2...102, in total 101 validators;
6. No validator gets a quorum of validations for the same ledger (80% of 101) and no validator fully validates any ledger;
7. The Preferred Branch step is meant to help with this situation, by allowing validators to "switch branch."
8. Nodes 1...51 (which use UNL X), cannot "switch branch" to L' as they get more than 50% of validations (51 out of 101) for a descendant of L;
9. Nodes 52...102 (which use UNL Y), cannot "switch branch" to L as they get more than 50% of validations (51 out of 101) for a descendant of L';
10. "The network cannot ever join without manual intervention" [5], i.e., it halts.

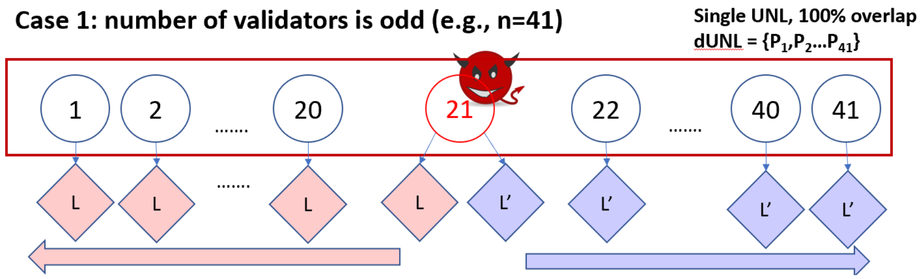

Figure 5: Attack by a single Byzantine validator with 100% UNL overlap.

B.3 Liveness Violation with 100% UNL Overlap and Single Byzantine Validator

Beyond their Example 9 illustrated in the previous section, Chase and MacBrough further argue (Theorem 11, [5]) that XRP Ledger Consensus Protocol guarantees liveness in case UNL overlap is 100%.

This is incorrect, as their analysis assumes "Byzantine accountability", i.e., limitations in potential misbehavior of Byzantine nodes which disallows a simple and standard attack by Byzantine validators in which Byzantine validators provide different information to different correct validators.

Refuting this claim of Chase and MacBrouh, we show that the XRP Ledger Consensus Protocol fails to guarantee liveness, even with 100% overlap across all UNLs, if one validator in the common UNL can be Byzantine (malicious) and if the network is unreliable.[39]

Consider the following example, which resembles Example 9 of [5] we depicted in Appendix B.2.

In this example there is a single UNL (100% overlap), and one Byzantine validator. The example uses 41 validators, as this is currently the actual number of validators in the dUNL in the XRP Ledger network, since July 16, 2021. The example is illustrated in Figure 5, only slightly modifies the Example 9 of [5] and goes as follows:

1. 41 validators experiencing Network Split in Deliberation;
2. Validators 1…20 send validation for descendant of ledger L;
3. Validators 22…41 send validation for descendant of ledger L';
4. Validator 21 is Byzantine, it sends validation for descendant of L to validators 1…20 and validation for descendant of L' to validators 22…41.
5. There is a single UNL, dUNL, containing all 41 validators.
6. No validator gets a quorum of validations for the same ledger (80% of 41) and no validator fully validates any ledger;
7. The Preferred Branch step is meant to help with this situation, by allowing validators to "switch branch."

[39] Our argument is similar to, but in its essence different from, the one presented by Amores-Sesar et al. [25] to which a short rebuttal was written by Ripple's employee Ethan Macbrough, as seen in the Twitter thread at `https://twitter.com/cczurich/status/1334153938241720322` and replies therein.

8. Nodes $1 \ldots 20$ cannot "switch branch" to L' as they get more than 50% of validations (21 out of 41) for descendant of L;
9. Nodes $22 \ldots 41$ cannot "switch branch" to L as they get more than 50% of validations (21 out of 41) for descendant of L';
10. "The network cannot ever join without manual intervention", i.e., it halts.